

# Assignment Submission

Lanston Chen

November 4, 2023

# Contents

|          |   |          |
|----------|---|----------|
| <b>1</b> | <b>Submission Details</b>   | <b>3</b> |
| 1.1      | RDS Endpoint and Table Names . . . . .                            | 3        |
| 1.1.1    | RDS code (create table / load data/ create primary key) . . . . . | 3        |
| 1.2      | List of Files in S3 Bucket . . . . .                              | 6        |
| 1.3      | List of Files in Hadoop Folder . . . . .                          | 7        |
| 1.4      | Oozie Workflow Completion or XML File . . . . .                   | 8        |

# 1 Submission Details

## 1.1 RDS Endpoint and Table Names

RDS endpoint: `database-lab5.c3f4g2yxdyve.us-east-1.rds.amazonaws.com`

List of table names:

- happy2015
- happy2016
- happy2017
- happy2018
- happy2019

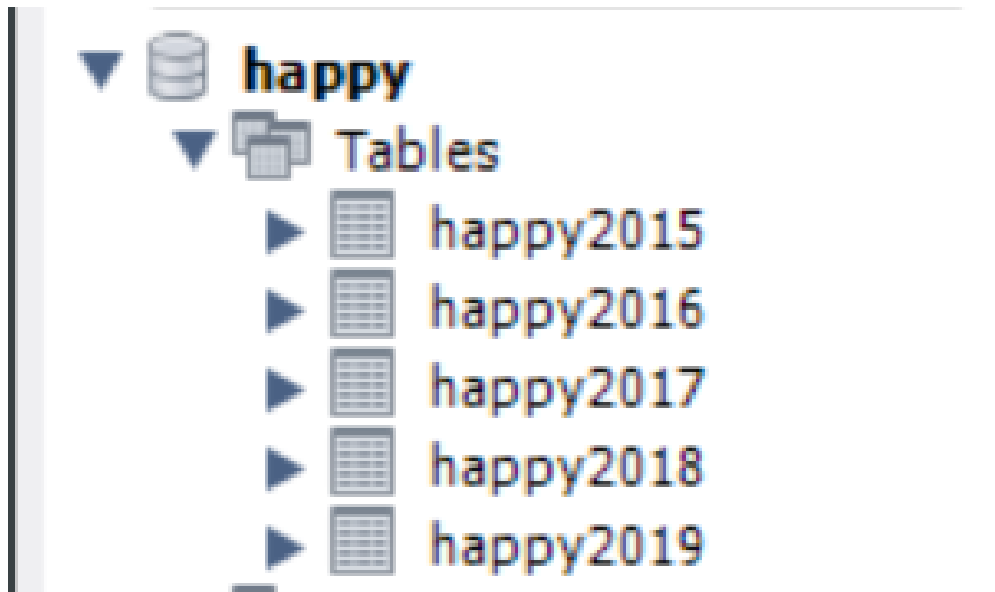


Figure 1: Table Name

### 1.1.1 RDS code (create table / load data/ create primary key)

```
literate
import os
import pandas as pd
from sqlalchemy import create_engine

# Database credentials
host = "database-lab5.c3f4g2yxdyve.us-east-1.rds.amazonaws.com"
port = "3306"
user = "admin"
```

```

password = "Ct123456"
database = "happy"

# Directory containing CSV files
directory = "C:\\Users\\ctlan\\OneDrive\\desktop\\manage_
            bigdata\\assignment\\HW\\hw5\\worldhappiness\\"

# Create a connection to the database
engine =
    create_engine(f'mysql+mysqlconnector://{user}:{password}@{host}:{port}/{database}')

# Define a dictionary for the year and its corresponding happiness score
column name
happiness_score_columns = {
    '2015.csv': 'Happiness_Score',
    '2016.csv': 'Happiness_Score',
    '2017.csv': 'Happiness.Score',
    '2018.csv': 'Score',
    '2019.csv': 'Score'
}

# Process each CSV file specified in the dictionary
for filename, happiness_column in happiness_score_columns.items():
    file_path = os.path.join(directory, filename)
    table_name = str("happy") + (str(os.path.splitext(filename)[0])) # Use
    filename without '.csv' as table name

    # Load the CSV file into a pandas DataFrame
    df = pd.read_csv(file_path)

    # Standardize column name to 'Happiness Score'
    df.rename(columns={happiness_column: 'Happiness_Score'}, inplace=True)

    # Check if 'Country' or 'Country or region' is the column name and
    standardize
    if 'Country' in df.columns:
        country_column = 'Country'
    elif 'Country_or_region' in df.columns:
        country_column = 'Country_or_region'
    else:
        raise ValueError('Country_column_not_found_in_the_file' + filename)

    # Sort by 'Happiness Score' and 'Country', reset index to create 'id'
    column

```

```

df.sort_values(by=['Happiness_Score', country_column],
ascending=[False, True], inplace=True)
df.reset_index(drop=True, inplace=True)
df.index += 1 # Make 'id' start from 1 instead of 0
df.reset_index(inplace=True)
df.rename(columns={'index': 'id'}, inplace=True)

# Create or replace the table and load data
df.to_sql(name=table_name, con=engine, if_exists='replace',
index=False, chunksize=500)

print(f"Data from {file_path} has been loaded into the {table_name}
table.")

print("All specified CSV files have been processed and corresponding tables
created.")
\end{listing}

\subsection{S3 Bucket URI}
S3 bucket URI: \texttt{s3://lanston-lb5-happy671/671data/happy\_2018 }

\subsection{EMR Master Public DNS}
EMR master public DNS for SSH:
\texttt{ec2-54-162-119-106.compute-1.amazonaws.com}

\subsection{EMR Hue Application Interface Link}
EMR Hue interface link:
\href{http://ec2-54-197-65-78.compute-1.amazonaws.com:8888/hue/oozie/editor/workflow/
Website}
{workflow=18}

\subsection{Sqoop Command Executions}
\begin{listing}[language=bash]
sqoop import \
--connect
    jdbc:mysql://database-lab5.c3f4g2yxdyve.us-east-1.rds.amazonaws.com/happy
\
--username admin \
--password Ct123456 \
--table happy2018 \
--target-dir /user/hadoop/lab5 \
--split-by id

```

```

2023-11-02 22:40:08,016 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1698963551667_0002
2023-11-02 22:40:08,016 INFO mapreduce.JobSubmitter: Executing with tokens: []
2023-11-02 22:40:08,220 INFO conf.Configuration: resource-types.xml not found
2023-11-02 22:40:08,221 INFO resource.ResourceUtiliti: Unable to find 'resource-types.xml'.
2023-11-02 22:40:08,442 INFO impl.YarnClientImpl: Submitted application application_1698963551667_0002
2023-11-02 22:40:08,887 INFO mapreduce.Job: The url to track the job: http://ip-172-31-20-106.ec2.internal:20888/proxy/application_1698963551667_0002/
2023-11-02 22:40:08,887 INFO mapreduce.Job: Running job: job_1698963551667_0002
2023-11-02 22:40:16,990 INFO mapreduce.Job: Job job_1698963551667_0002 running in uber mode : false
2023-11-02 22:40:16,990 INFO mapreduce.Job: map 0% reduce 0%
2023-11-02 22:40:23,180 INFO mapreduce.Job: map 50% reduce 0%
2023-11-02 22:40:25,189 INFO mapreduce.Job: map 100% reduce 0%
2023-11-02 22:40:25,196 INFO mapreduce.Job: Job job_1698963551667_0002 completed successfully
2023-11-02 22:40:25,306 INFO mapreduce.Job: Counters: 33
  File System Counters
    FILE: Number of bytes read=0
    FILE: Number of bytes written=1196185
    FILE: Number of read operations=0
    FILE: Number of large read operations=0
    FILE: Number of write operations=0
    HDFS: Number of bytes read=395
    HDFS: Number of bytes written=886
    HDFS: Number of read operations=24
    HDFS: Number of large read operations=0
    HDFS: Number of write operations=8
    HDFS: Number of bytes read erasure-coded=0
  Job Counters
    Launched map tasks=4
    Other local map tasks=4
    Total time spent by all maps in occupied slots (ms)=1056976
    Total time spent by all reducers in occupied slots (ms)=0
    Total time spent by all map tasks (ms)=17048
    Total vcore-milliseconds taken by all map tasks=33823232
    Total megabyte-milliseconds taken by all map tasks=33823232
  Map-Reduce Framework
    Map input records=156
    Map output records=156
    Input split bytes=395
    Spilled Records=0
    Failed Shuffles=0
    Merged Map outputs=0
    GC time elapsed (ms)=630
    CPU time spent (ms)=12440
    Physical memory (bytes) snapshot=2022166528
    Virtual memory (bytes) snapshot=14659862528
    Total committed heap usage (bytes)=6383730688
    Peak Map Physical memory (bytes)=51061456
    Peak Map Virtual memory (bytes)=3674796032
  File Input Format Counters
    Bytes Read=0
  File Output Format Counters
    Bytes Written=886
2023-11-02 22:40:25,310 INFO mapreduce.ImportJobBase: Transferred 8.6777 KB in 21.7122 seconds (409.2621 bytes/sec)
2023-11-02 22:40:25,312 INFO mapreduce.ImportJobBase: Retrieved 156 records.

```

Figure 2: Sqoop Command Executions

## 1.2 List of Files in S3 Bucket

```

2023-11-02 22:40:08,016 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1698963551667_0002
2023-11-02 22:40:08,016 INFO mapreduce.JobSubmitter: Executing with tokens: []
2023-11-02 22:40:08,220 INFO conf.Configuration: resource-types.xml not found
2023-11-02 22:40:08,221 INFO resource.ResourceUtiliti: Unable to find 'resource-types.xml'.
2023-11-02 22:40:08,442 INFO impl.YarnClientImpl: Submitted application application_1698963551667_0002
2023-11-02 22:40:08,887 INFO mapreduce.Job: The url to track the job: http://ip-172-31-20-106.ec2.internal:20888/proxy/application_1698963551667_0002/
2023-11-02 22:40:08,887 INFO mapreduce.Job: Running job: job_1698963551667_0002
2023-11-02 22:40:16,990 INFO mapreduce.Job: Job job_1698963551667_0002 running in uber mode : false
2023-11-02 22:40:16,990 INFO mapreduce.Job: map 0% reduce 0%
2023-11-02 22:40:23,180 INFO mapreduce.Job: map 50% reduce 0%
2023-11-02 22:40:25,189 INFO mapreduce.Job: map 100% reduce 0%
2023-11-02 22:40:25,196 INFO mapreduce.Job: Job job_1698963551667_0002 completed successfully
2023-11-02 22:40:25,306 INFO mapreduce.Job: Counters: 33
  File System Counters
    FILE: Number of bytes read=0
    FILE: Number of bytes written=1196185
    FILE: Number of read operations=0
    FILE: Number of large read operations=0
    FILE: Number of write operations=0
    HDFS: Number of bytes read=395
    HDFS: Number of bytes written=886
    HDFS: Number of read operations=24
    HDFS: Number of large read operations=0
    HDFS: Number of write operations=8
    HDFS: Number of bytes read erasure-coded=0
  Job Counters
    Launched map tasks=4
    Other local map tasks=4
    Total time spent by all maps in occupied slots (ms)=1056976
    Total time spent by all reducers in occupied slots (ms)=0
    Total time spent by all map tasks (ms)=17048
    Total vcore-milliseconds taken by all map tasks=33823232
    Total megabyte-milliseconds taken by all map tasks=33823232
  Map-Reduce Framework
    Map input records=156
    Map output records=156
    Input split bytes=395
    Spilled Records=0
    Failed Shuffles=0
    Merged Map outputs=0
    GC time elapsed (ms)=630
    CPU time spent (ms)=12440
    Physical memory (bytes) snapshot=2022166528
    Virtual memory (bytes) snapshot=14659862528
    Total committed heap usage (bytes)=6383730688
    Peak Map Physical memory (bytes)=51061456
    Peak Map Virtual memory (bytes)=3674796032
  File Input Format Counters
    Bytes Read=0
  File Output Format Counters
    Bytes Written=886
2023-11-02 22:40:25,310 INFO mapreduce.ImportJobBase: Transferred 8.6777 KB in 21.7122 seconds (409.2621 bytes/sec)
2023-11-02 22:40:25,312 INFO mapreduce.ImportJobBase: Retrieved 156 records.

```

Figure 3: S3 Bucket

### 1.3 List of Files in Hadoop Folder

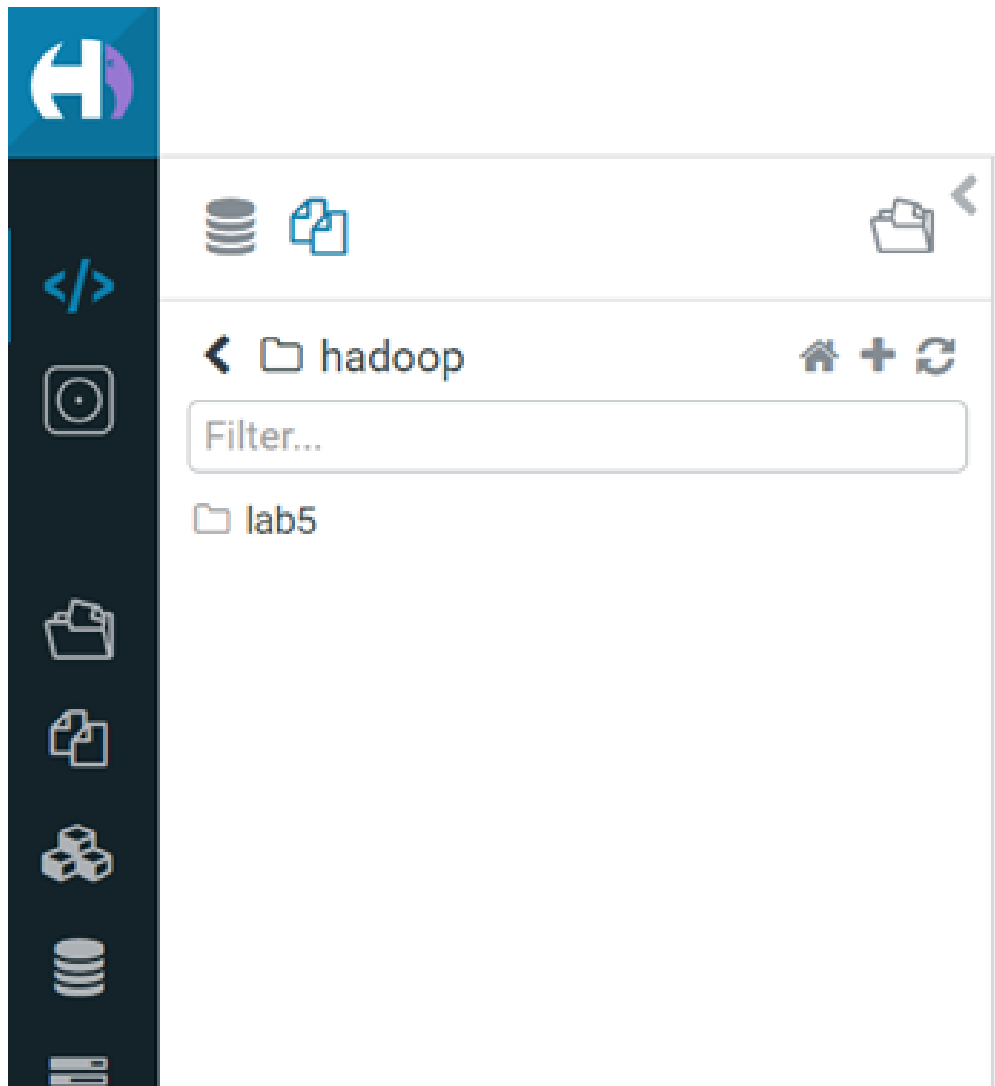


Figure 4: Hadoop Folder

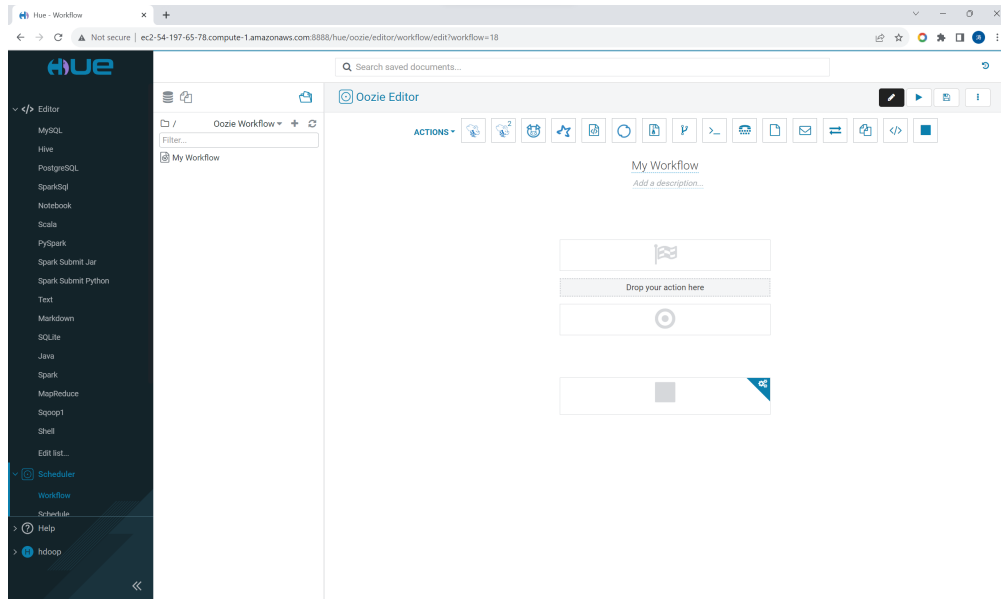


Figure 5: Hadoop Interface

## 1.4 Oozie Workflow Completion or XML File

```
<!-- <workflow-app name="Lab 5 Workflow" xmlns="uri:oozie:workflow:0.5">
  <start to="sqoop-71ff"/>
  <kill name="Kill">
    <message>Action failed, error message[\$\{wf:errorMessage(wf:lastErrorNode())\}]
  </kill>
  <action name="sqoop-71ff">
    <sqoop xmlns="uri:oozie:sqoop-action:0.2">
      <job-tracker>\$\{jobTracker\}</job-tracker>
      <name-node>\$\{nameNode\}</name-node>
      <command>import --connect jdbc:mysql://database-lab5.c3f4g2yxdyve.us-east-1.
    </sqoop>
    <ok to="sqoop-7711"/>
    <error to="Kill"/>
  </action>
  <action name="sqoop-7711">
    <sqoop xmlns="uri:oozie:sqoop-action:0.2">
      <job-tracker>\$\{jobTracker\}</job-tracker>
      <name-node>\$\{nameNode\}</name-node>
      <command>import --connect jdbc:mysql://database-lab5.c3f4g2yxdyve.us-east-1.
    </sqoop>
    <ok to="email-98f8"/>
    <error to="Kill"/>
  </action>
  <action name="email-98f8">
    <email xmlns="uri:oozie:email-action:0.2">
```



```
<to>tche368@emory.edu</to>
<subject>Workflow Notification: Import Complete</subject>
<body>The Sqoop import actions have completed successfully.</body>
<content_type>text/plain</content_type>
</email>
<ok to="End"/>
<error to="Kill"/>
</action>
<end name="End"/>
</workflow-app>
-->
```